

Loglan'82

fundamental research

Andrzej Salwicki, editor

14 marca 2014

Contents

Preface	v
I Contributions	1
1 How to retain Display Vector?	5
2 Determining base class	7
3 Safe and efficient programmed deallocation of objects	9
4 Coroutines	11
5 Processes and their active objects	13
6 Alien call of methods of active objects	15
II Axioms	17
7 Syntax	21
8 Axioms and inference rules	23
9 Do we described the semantics?	25
10 Max - a mathematical model of concurrent computations	27
III Virtual machine	29

Preface

W tej książce przedstawiamy przegląd prac badawczych wykonanych w związku z projektem Loglan'82. Celem projektu było zaprojektowanie języka do programowania obiektowego i rozproszonego i zbudowanie kompilatora tego języka na maszyny MERA-400 produkowane w Polsce w latach 1970. Sam język nie pozyskał zwolenników. Istnieje wersja kompilatora działająca na platformach Linux i Windows. Można też przenieść kompilator na klastry np.

Chcemy zwrócić uwagę na wyniki prac o charakterze badań podstawowych jakie poprzedziły powstanie języka Loglan i jego kompilatora.

Pytania jakie sobie stawialiśmy to:

- Czy język programowania, który zezwala na *zagnieżdżanie* modułów i na ich *rozszerzanie* (tj. dziedziczenie) może wykorzystywać efektywny mechanizm dostępu do zmiennych jakim jest Display Vector zaproponowany przez E.W.D. Dijkstrę []?
- W deklaracji klasy A występuje nazwa klasy dziedziczonej B. Program może zawierać wiele klas o nazwie B. W jaki sposób wyznaczyć tę właściwą klasę B? Problem ten staje się jeszcze bardziej złożony w Javie. Mamy nadzieję, że nasze wyniki mogą się przydać twórcom kolejnych kmpilatorów Javy.
- Zarządzanie pamięcią obiektów to nawigacja pomiędzy dwoma potworami: *wyciekami* pamięci i *wiszącymi referencjami*. Antoniemu Kreczmarowi zawdzięczamy rozwiązanie bezpieczne i efektywne pozwalające programiście usuwać obiekty bez obawy o pozostawienie referencji do usuniętego obiektu.
- Współprogramy (*ang. coroutine*) to narzędzie programowania znane od 60 lat ale zapomniane. Dziś coraz częściej twórcy nowych języków programowania przypominają je, ambitni programiści odkrywają zalety programowania quasi-współbieżnego. Wprowadzenie współprogramów

do Loglanu zostało poprzedzone ich analizą. Uwolniliśmy się od niektórych błędnych założeń, jakie występują np. w języku Simula67.

- Koncepcja obiektu klasy jaką przyjęto w Loglanie'82 różni się od definicji przyjętych w językach C+, Java i in.
- Moduły procesów pozwalają tworzyć wątki, a dokładniej obiekty aktywne. Tworząc obiekt aktywny możemy go ulokować na tym samym komputerze, który opracowuje polecenie
 $x := \text{new MojProces}(\dots)$
 lub na innym komputerze uprzednio dołączonym siecią. Loglan wykonywany jest na wirtualnym procesorze Loglanu - VLP. Procesy takie można łączyć i tworzyć w ten sposób klaster - wieloprocessorowy wirtualny komputer loglanowski WWKL.
- Loglan'82 oferuje oryginalny sposób synchronizacji/komunikacji pomiędzy obiektami aktywnymi procesów. Jest to tzw. obce wołanie metod (*ang.* alien call).
- czy tylko templety są rozwiązaniem problemu przekazywania typów jako parametrów

This book presents a survey of fundamental work done for Loglan'82 project.

Programming languages are an important tool in information technology. Dlaczego?

Loglan is a programming language that appeared in late 70 of XXth century.

Part I

Problems and solutions

This part contains exposition of six problems encountered during work on design of Loglan'82. that a designer of a programming language

Chapter 1

How to retain Display Vector?

One of first concerns that the designers of Loglan had to cope with, was how to keep in use the mechanism of Display Vector.

3 papers are devoted to the problem:

- Antoni Kreczmar, W.M. Bartol, H. Oktaba, A.I. Litwiniuk: *Semantics and Implementation of Prefixing at Many Levels*. in Proc. Logics of Programs and their Applications Symposium Poznań 1980. Berlin, Springer Vlg, 1983, pp. 45-80, series: LNCS 148, ISBN 0387119817 [2]
- Antoni Kreczmar, Manfred Krause, Hans Langmaack, Andrzej Salwicki: *Specification and Implementation Problems of Programming Languages Proper for Hierarchical Data Types*. T. Rep. 8410. Kiel: Institut fuer Informatik, University of Kiel, 1984.
- Antoni Kreczmar, Manfred Krause, Hans Langmaack, Marek Warpechowski: *Concatenation of program modules an algebraic approach to the semantics and implementation problems*. Berlin: Springer, 1986, s. 134-156, seria: LNCS 208.

Chapter 2

Determining base class

When a programmer writes:

`class A extends class B`, or in Java, `class A extends B.C.D`

it means that name B should be replaced by a unique class of the name D

The problem was solved for Loglan by Danuta Szczepańska.[see DSW.f file of the compiler] In Java the situation became much more difficult.

2 papers

Chapter 3

Safe and efficient programmed deallocation of objects

Antoni Kreczmar solved the problem of safe, programmed deallocation of objects in its running system of Loglan'82 programming language.

Later in a joint paper with Gianna Cioni he described the most important parts of the subsystem of management of objects [1]. A detailed study of specification of Kreczmar's system was done by Hanna Oktaba in her ph.d. thesis

In a recent paper we compare the the most popular programming languages with respect to the problems of memory leakage and deallocation of objects.

Chapter 4

Coroutines

Antoni Kreczmar proposed the system of coroutines which is simpler than one proposed in Simula67 and it is free of inconsistency of the Simula.

Chapter 5

Processes and their active objects

In 1988 Bolek Ciesielski presented

Chapter 6

Alien call of methods of active objects

The fundamental concept of Bolek Ciesielski proposal is an **alien call** concept.

Part II

Axiomatic semantics of a sublanguage

In this part we analyze semantics of basic constructs of Loglan'82. The results presented here belong to another research project - Algorithmic Logic. We include them here for it is first time that semantics is subsumed in a few lines through axioms and inference rules.

It should be obvious that this part reports a not yet finished research.

Chapter 7

Syntax

Chapter 8

Axioms and inference rules

Chapter 9

Do we described the semantics?

Chapter 10

Max - a mathematical model of concurrent computations

It is widely accepted that interleaving of atomic actions of processes is a model of concurrent computations which precisely corresponds to possible behaviours of a concurrent program. We have invented another model. F de roever called it a true concurrency model.

Powiedz trochę o modelowaniu obliczeń przez sieci Petriego. Opowiedz o L. Lamporta pomysle na zbudowanie maszyny, która ma tylko sprawiedliwe (fair) obliczenia.

Part III

Virtual machine

The running system for Loglan'82 was originally written by Antoni Kreczmar in assembler for MERA400 computers. Next, Paweł Gburzyński made a virtual machine in Pascal. It was a part of the project: *port of Loglan'82 onto mainframe plate-form*. This was done in Kiel University in 1984. Later, the virtual machine was translated to C language by G. Grudziński and M. Benke, in this way the ports onto PC DOS plate-form, Linux plateform and VAX/VMS plate-form were done.

Antoni Kreczmar left us a version of runnning system (today it is called a virtual machine) of next version of Loglan language. Unfortunately, the compiler for this version does not exist.

Bibliography

- [1] G. Cioni and A. Kreczmar. Programmed Deallocation without Dangling References. *Information Processing Letters*, 18:179–185, 1984.
- [2] Oktaba H. Ratajczak W.M. Litwiniuk A.I. Kreczmar, A. Semantics and Implementation of Prefixing at Many Levels. In A. Salwicki, editor, *Logics of Programs and their Applications*, *LNCS 148*, pages 45 –80. Springer Vlg, 1983.